BASICS of react routing with react-router-dom

(Only this first basic part asked in the exam)

1. Install react-router-dom

https://reactrouter.com/en/v6.3.0/getting-started/overview#installation

npm install react-router-dom (BTW. version 6 going on)

2. Wrap your root component inside e.g. BrowserRouter

https://reactrouter.com/en/v6.3.0/getting-started/overview#configuring-routes

3. Define the mappings from a route="frontend url" to component

https://reactrouter.com/en/v6.3.0/getting-started/overview#configuring-routes (same code example)

4. Add clickable links for user to be able to use routing

https://reactrouter.com/en/v6.3.0/getting-started/tutorial#add-some-links

5. ... those can have routing parameters

https://reactrouter.com/en/v6.3.0/getting-started/overview#reading-url-parameters

(why on earth they call it URL? It's inside of React-router DOM routing in frontend, not internet nor backend)

let params = useParams();

const categoryId = params.categoryId;

Not in exam:

But really useful **new features** for making frontend development that makes sense

6. Nested routing would be a new option, helps in organizing app to different areas, e.g. student's view navigation

and school admins views' navigation

https://reactrouter.com/en/v6.3.0/getting-started/overview#nested-routes

7. Nested routing

+ Layout (common layout "frame" under this! Thus not only one layout frame for whole app!)

+ Outlet (place to fill by individual "child components")

+ Index (e.g. the default child components to go to)

https://reactrouter.com/en/v6.3.0/getting-started/overview#index-routes

8. +Relative links, easier maintenability

https://reactrouter.com/en/v6.3.0/getting-started/overview#relative-links

Relative <Link to> values.

9. 404 routes with "*" at the end

https://reactrouter.com/en/v6.3.0/getting-started/overview#not-found-routes

10. Multiple different, independent routing sections in our SPA

https://reactrouter.com/en/v6.3.0/getting-started/overview#multiple-sets-of-routes

11. Allowed to add more <Route>:s in child components, not just in one place in the app.

(Recommended only if you have a very good plan? Otherwise can be messy to follow?)

https://reactrouter.com/en/v6.3.0/getting-started/overview#descendant-routes

12. Active link showing where user is right now, in the navigation link list

https://reactrouter.com/en/v6.3.0/getting-started/tutorial#active-links

13. useOutletContext - Now you can have the subsection of your app having shared objects!

https://reactrouter.com/en/main/hooks/use-outlet-context